Engineering Science and Technology, an International Journal 17 (2014) 165–172

HOSTED BY

Contents lists available at ScienceDirect

Engineering Science and Technology, an International Journal

journal homepage: http://www.elsevier.com/locate/jestch

Full length article

A novel MLP network implementation in CMOL technology

Abdalhossein Rezai [a,b,*], Parviz Keshavarzi [a], Reza Mahdiye [a]

[a] Electrical and Computer Engineering Faculty, Semnan University, 3513119111 Semnan, Iran
[b] Academic Center for Education, Culture and Research (ACECR), Isfahan University of Technology (IUT) Branch, Isfahan, Iran

ARTICLE INFO

Article history:
Received 1 February 2014
Received in revised form
30 April 2014
Accepted 30 April 2014
Available online 6 June 2014

Keywords:
CMOL technology
Nanotechnology
Nanoelectronics
Nanodevice
MLP network

ABSTRACT

Hybrid CMOS/nanodevice technology is a well-known candidate to extend the exponential Moor-Law progress of microelectronics beyond the 10-nm frontier. This paper presents and evaluates a novel method for synaptic weights implementation of artificial neural networks in CMOL technology, a hybrid CMOS/nanodevice technology. In this novel method, the analog property of the I–V characteristic of the nanodevice is utilized to implement each neuromorphic synaptic weight. Each synaptic weight is also implemented by using one nanodevice instead of several nanodevices. Moreover, the proposed method is applied to the multilayer perceptron (MLP) network in CMOL technology. Our analysis shows that the power consumption and speed are effectively improved in the proposed method compared to other methods at the expense of a reasonable overhead defect tolerance.

1. Introduction

Recently, extensive research attempts have been carried out on artificial neural network (ANN) implementations and applications [5,11,16,19,28,42]. Since reducing the implementation area and power consumption are very important issues in VLSI designs, nanotechnology is a well-known candidate to develop these implementations [15,22,37].

In the last decade, hybrid CMOS/nanodevice technologies such as CMOL, and Field-Programmable Nanowire Interconnect (FPNI), have received widespread attention due to the number of promising applications [29]. These applications can be classified into three important groups: (1) design of digital and analog circuits [3,10,29,30], (2) design of nonvolatile memories [13,25,32,43], and (3) design of artificial neural networks [8,18,21,22,34,37–39].

In these technologies, a CMOS subsystem with relatively large silicon transistors is utilized for signal restoration, long-range communications, input/output functions, and testing, while add-on nanowire crossbar with simple two-terminal nanodevices at each Crosspoint provides most of the information storage, short-range communications, and facilitate signal processing [8,15,22,34,36].

Artificial neural network is a candidate to be developed in hybrid CMOS/nanodevice technologies such as CMOL due to the ultra-dense circuit integration demonstrated in these technologies [8,15].

In neurocomputing, the setting of synapses weight is an important issue [15]. The use of two-terminal nanodevices to implement synaptic weights is investigated in literature [8,15]. Likharev [21,22], and Turel and Likharev [39] have introduced the CrossNets such as Inbar, Flossbar and Randbar. Moreover, Afifi et al. [1], Lee and Likharev [20], Likharev [23], Likharev et al. [24], and Turel et al. [40,41] have investigated the neural network implementations such as multilayer perceptron (MLP) and Hopfield in CMOL technology.

Folling et al. [8], Likharev [21,22], and Turel et al. [41] have shown that each synaptic weight can implement using a square array of $n \times n$ nanodevices. In this method, the number of on-state nanodevices determines the synaptic weight. In other words, the synaptic weight can be controlled by changing the number of on-state nanodevices.

This paper investigates a novel method to implement the synaptic weight in the artificial neural network in CMOL technology. In this novel method, each synaptic weight is implemented by only one nanodevice based on the analog property of its I–V characteristic. The proposed method is also applied to the MLP network in CMOL technology. The MNIST database [27] is utilized to training and testing of the proposed method. Our analysis shows that the proposed method for the MLP network implementations has several advantages over other methods for the MLP network

* Corresponding author. Academic Center for Education, Culture and Research (ACECR), Isfahan University of Technology (IUT) Branch, Isfahan, Iran.
E-mail addresses: rezaie@acecr.ac.ir, rezai560516@gmail.com (A. Rezai), pkeshavarzi@semnan.ac.ir (P. Keshavarzi), r_mahdiye@semnan.ac.ir (R. Mahdiye).
Peer review under responsibility of Karabuk University.

http://dx.doi.org/10.1016/j.jestch.2014.04.009

implementations such as Folling et al. [8], Likharev [21,22], and Turel et al. [41].

The rest of this paper is organized as follows: Section 2 describes CMOL technology, and CMOL-based artificial neural networks. The proposed method for the MLP network implementations is presented in Section 3. Section 4 evaluates the proposed method. Finally, Section 5 concludes the paper.

2. Background

2.1. CMOL technology

One of the known ways to open new opportunities for VLSI designs is to utilize hybrid CMOS/nanodevice technologies [7,12,29,37]. Fig. 1 shows the general idea of hybrid CMOS/nanodevice technology.

The implementation platform in the hybrid CMOS/nanodevice technologies combines a typical CMOS subsystem, with bottom layer of silicon MOSFETs and several wiring layers, augmented with a simple nanoelectronic add-on layer. The nanoelectronic layer consists of a Crossbar of nanowires with two-terminal nanodevices placed in Crosspoints of nanowires [29,37].

CMOL is a well-known example of hybrid CMOS/nanodevice technologies. In CMOL technology, the most difficult functions are moved into CMOS subsystem. The nanoelectronic layer, nanowires and two-terminal nanodevices, is utilized for wired-OR logic and signal routing [29,37]. Fig. 2 shows schematic diagram of CMOL technology.

In CMOL technology, a nanowire Crossbar is placed on top of a sea of CMOS inverters (Fig. 2b–d). Fig. 2a shows interface between CMOS subsystem and nanoelectronic layer, which is provided by pins. These pins are distributed all over the circuit area [23,34,37]. The Crossbar is slightly rotated so that each nanowire is electrically connected to one pin extending up from CMOS subsystem (Fig. 2a and b). Two-terminal nanodevices are assumed to be nonlinear antifuses to implement the wired-OR function, but device variability or insufficient nonlinearity might decrease the circuit density [34,37]. For example in Fig. 2d, if only two Crosspoint nanodevices, connected to inputs A and B, are in the on-state, the output shows NOR function of input signals A and B, $F = \overline{A + B}$. Different investigations show that the cleverness of this technology is its simplicity, density, and clean separation of configuration and data communication [15,23,29,37].

2.2. CMOL-based artificial neural networks

Artificial neural networks implemented in CMOL technology, similar to conventional neural networks, consist of neural cell bodies, or somas, synapses, axons and dendrites. These networks, which are also named CrossNet, using CMOS subsystem to implement the neural cell bodies, or somas, the nanodevices as synapses, and interconnects or nanowires as axons and dendrites. Somas are also connected through the nanowires and nanodevices [23,39,41]. Fig. 3 shows a simplified schematic of a CMOL-based artificial neural network.

The neural cell bodies, which are shown by gray squares in Fig. 3b, are always nonlinear amplifiers, which are typically implemented in CMOS subsystem. These amplifiers feed axons and are also fed by dendrites, which are shown by red and blue lines respectively in Fig. 3b, in their on-states [22,23,41]. Moreover, the direct connections between any two neurons located on the same row or column are excluded, because axonic and dendritic nanowires have open-circuit terminations on one end. As a result, neurons should communicate through the synapses.

In other words, output voltage of jth soma is applied to axonic nanowires, which are shown by red lines. Perpendicular, physically similar dendritic nanowires, which are shown by blue lines, lead to inputs of kth soma. When two-terminal nanodevice at the Crosspoint of jth axon and kth dendritic is in its on-state, this voltage provides a substantial contribution of the current injection I_k into kth dendritic nanowire. As a result, in the linear approximation and low input load, the total current is as follows:

$$I_j(t) = \sum_{k=1}^{m} \sum_{s_j = \pm 1} s_j G_{jk} V_k(t) \tag{1}$$

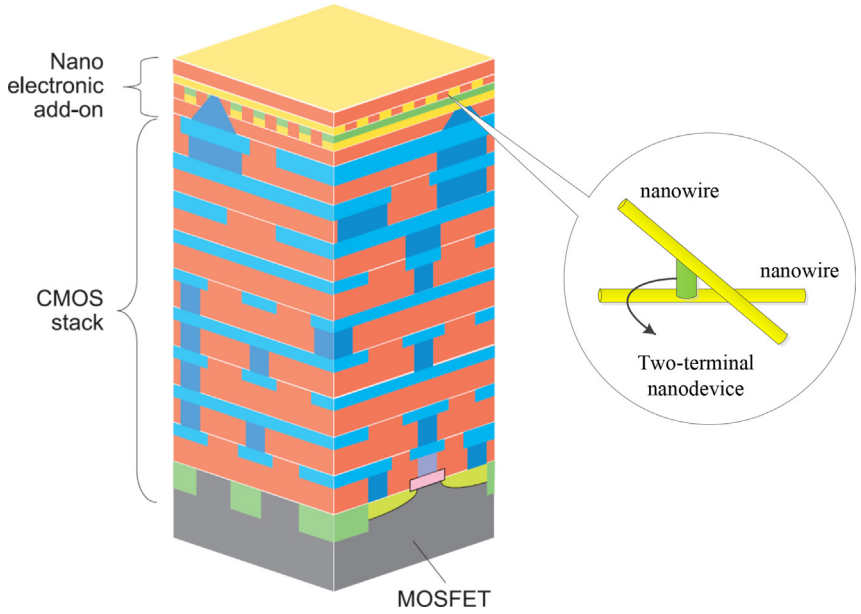


Fig. 1. The general idea of hybrid CMOS/nanodevice technology which consist of CMOS subsystem and nanoelectronic add-on layer. Two-terminal nanodevice is placed between two horizontal and vertical nanowires [29,37].

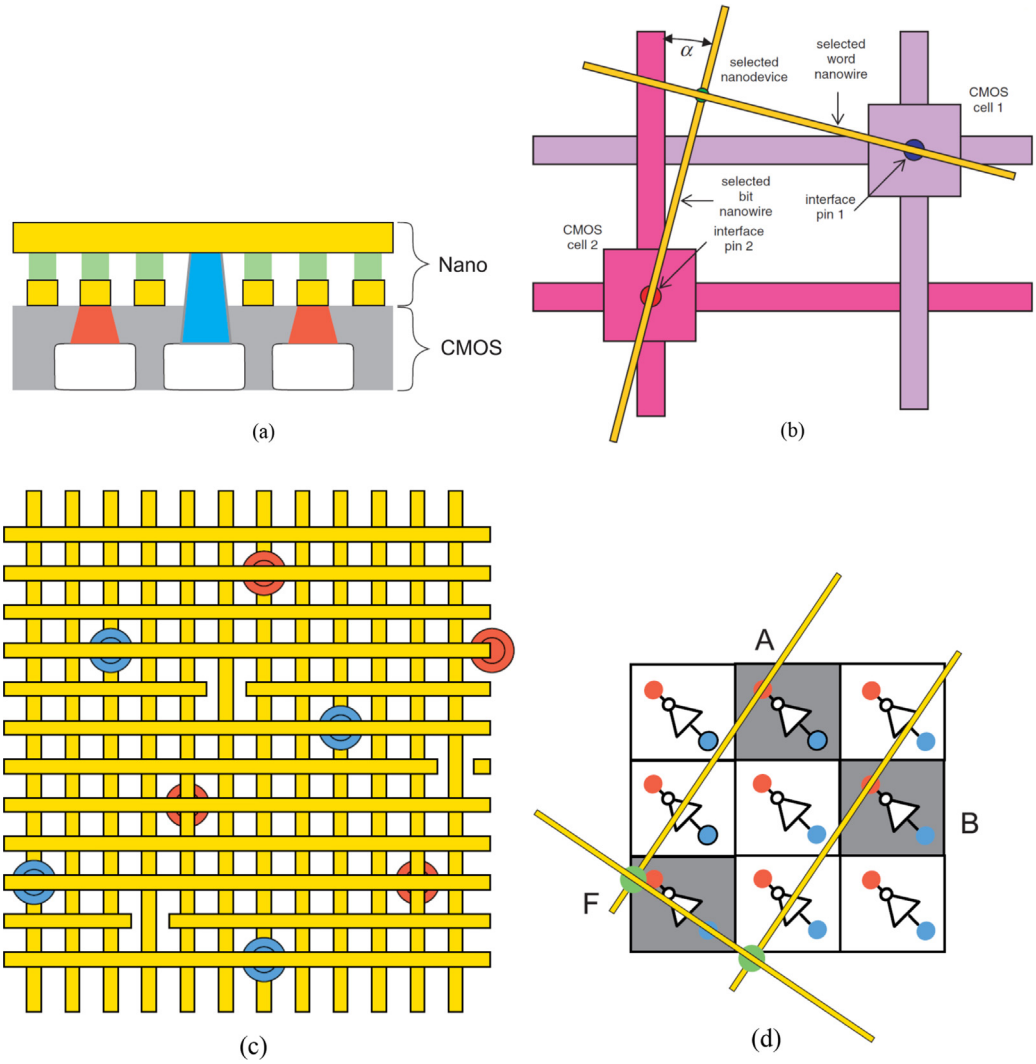


Fig. 2. Schematic diagram of CMOL technology (a) a schematic side view, (b) a schematic top view showing the idea of addressing a particular nanodevice via CMOS subsystem and interface pins, the specific rotation angle makes each nanowire individually accessible from CMOS subsystem, (c) a zoom-in top view on the circuit near several interface pins, (d) an example of 2-input NOR gate in which the output signal denotes the NOR of two input signals A and B, $F = \overline{A + B}$ [34,37].

where $s_j = \pm 1$ denotes the input polarity, G_{jk} denotes the conductance of the Crosspoint nanodevice, and $s_j G_{jk}$ plays the role of the synaptic weight [23,40]. The synaptic weight provided by a Crosspoint nanodevice is binary, while for most applications multi-valued synapses are required [37].

Folling et al. [8], Likharev [21,22], and Turel et al. [41] have shown that a square array of $n \times n$ nanodevices can be utilized to implement multi-valued synapses in the MLP network implementation. Fig. 4 shows a simplified schematic of this idea.

In this implementation method, analog addition of currents in a square array of $n \times n$ nanodevices produces a composite synapse with $n^2 + 1$ discrete weight levels [22]. Each level denotes the number of on-state nanodevices [23,41].

3. The proposed method for MLP network implementation

The MLP network implementation methods, which use a square array of $n \times n$ nanodevices for implementing each synaptic weight, have a long time and complex learning process. Moreover, the power consumption and area in these implementations are considerable. In these implementation methods, the digital properties of nanodevices are used (on-state and off-state), and the information loss at clipped synapse may seriously affect the network performance. It's expected that using one nanodevice for each synaptic weight will result in less power consumption and faster performance compared with multiple nanodevices. Therefore, we propose a new and efficient method for the MLP network implementations, which improves these conditions. The proposed method for the MLP network implementations is shown in Fig. 5.

In the proposed method for the MLP network implementation, axons and dendrites, which are shown by orange and blue lines respectively, are implemented as nanowires. Synapses, which are shown by green circles, are also implemented at Crosspoints. Neural cell bodies, somas, which are shown by rectangular, are implemented in CMOS subsystem. Moreover, the proposed method uses only one nanodevice to implement each synaptic weight. Any pair of somas is connected by two synapses leading to soma inputs. As a result, synaptic weight may effects as $-w$, 0, and $+w$ where w shows the synaptic weight.

In this method, the analog property of the I–V characteristic of the nanodevice is utilized to implement each synaptic weight. We utilized the suggested model by Hu et al. [14] for simulation. Simulation results of I–V characteristic of the nanodevice in on-

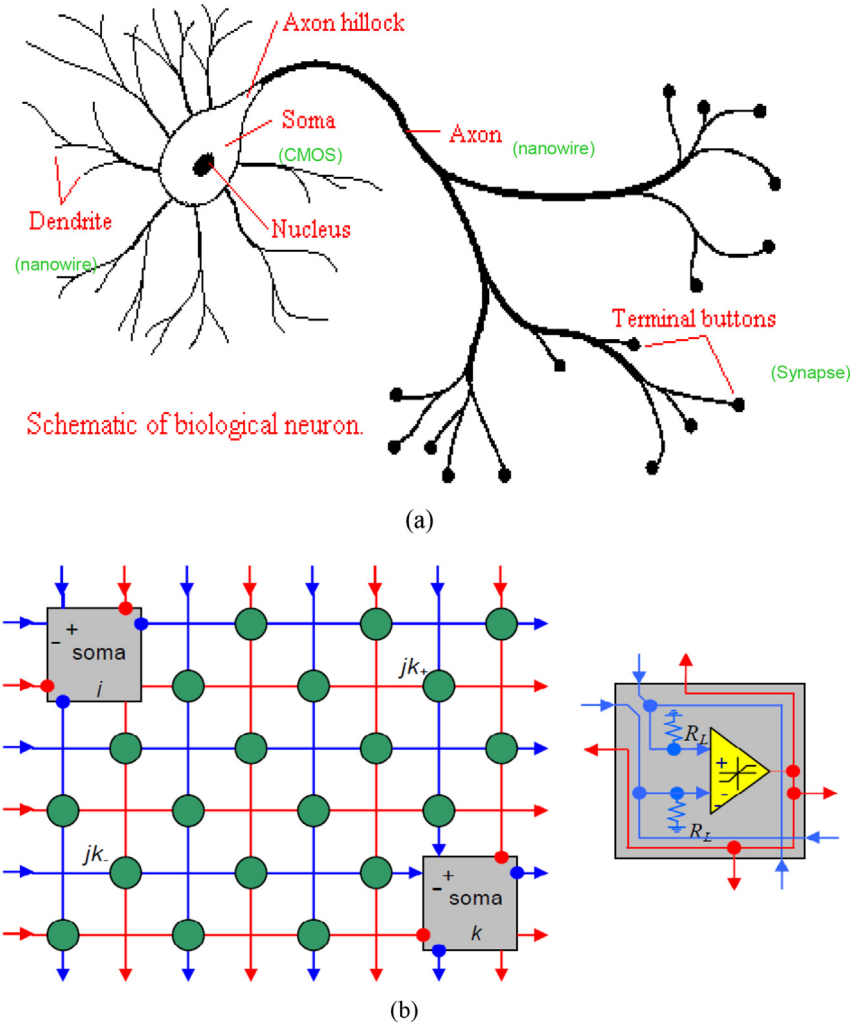


Fig. 3. A simplified schematic of a CMOL-based artificial neural network. a) Schematic of biological neuron, b) a simplified schematic of connection of two neurons and its soma for the non-adaptive, firing rate regime. Red and blue lines show axonic and dendritic nanowires respectively. Bold red and blue points show open-circuit terminals of nanowires, which do not allow somas to interaction bypass of synapses [22,41].

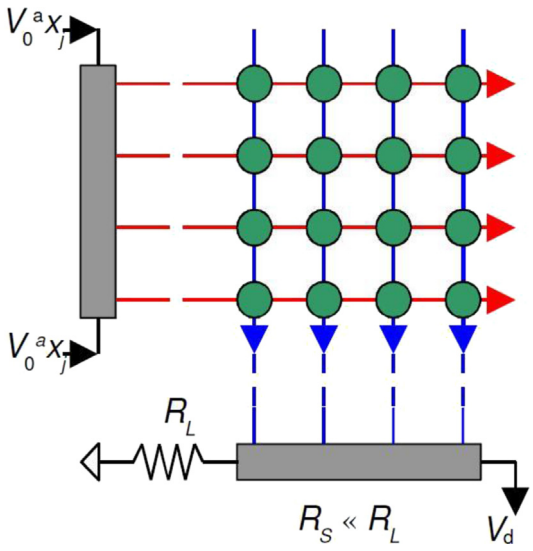


Fig. 4. The synaptic weights implementation method using a square array of $n \times n$ nanodevices. This method providing $n^2 + 1$ discrete weight levels [22].

state are illustrated in Fig. 6a. Fig. 6b shows Likharev [21] implementation results of the I–V characteristic of a nanodevice.

The nanodevice current uniquely depends on the voltage dropped across the nanodevice in the on-state. The I–V characteristic of the nanodevice plays an important role in the proposed method for the MLP network implementations.

In the proposed method, the training is done outside and the resulted weights are simply imported into the hardware. This procedure is started with training of a homomorphic precursor artificial neural network with synaptic weights, implemented in software, using one of established method such as error backpropagation. To implement weights, all two-terminal nanodevices are first reset to their off-state. Then, pairs of somatic cells are sequentially selected to apply external voltage to a particular synapse, so that the corresponding two-terminal nanodevice is turn on. Finally, for applying the desired voltage to the nanodevice, the following steps are performed:

1) The required nanodevice current is determined based on the synaptic weight. These two parameters have a linear relationship.
2) The required nanodevice voltage is determined based on the required nanodevice current and Fig. 6.
3) This desired voltage is applied to the nanodevice.

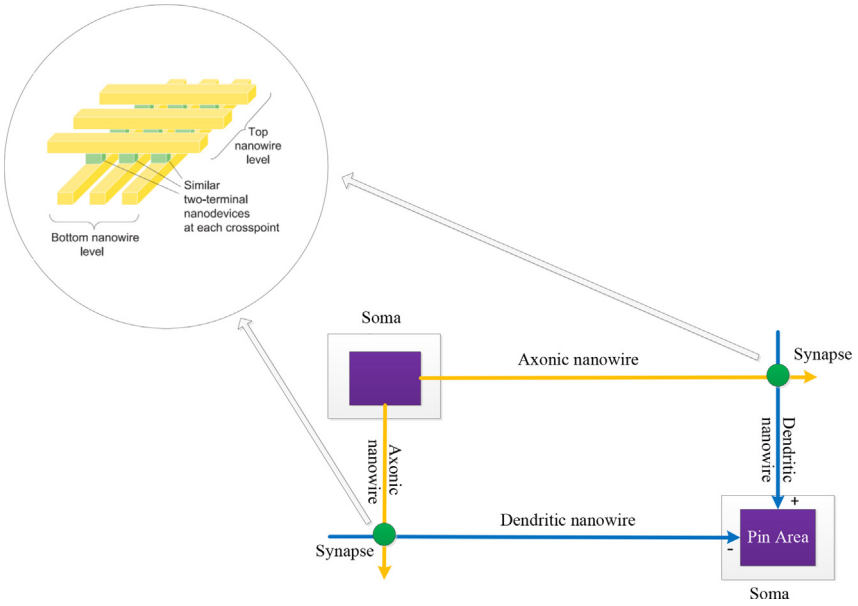


Fig. 5. The proposed method for the MLP network implementations. Axonic and dendritic nanowires are shown by orange and blue lines respectively; synapses implemented in Crosspoints are shown by green circles, and somas implemented in CMOS subsystem are shown by rectangular.

As a result, the CMOL-based artificial neural network can be dynamically trained in the proposed method for the MLP network implementations.

4. Results and discussions

In this section, the power consumption, speed and defect tolerance of the proposed MLP network implementation method are investigated. It should be noted that the components which can affect the performance of the implementations in CMOL technology are the number of required nanodevices, the nanowires, the pin-to-nanowire contact, and pins interface between the nanowires and the CMOS subsystem [9,26].

4.1. Power consumption evaluation

The static power consumption of the CMOL-based artificial neural networks includes both the working power and the leakage power [9]. The working power is based on the on-state conditions and is determined as follows [9]:

$$P_{on} = \alpha\beta NMV^2/(2R_{con} + R_{wire} + R_{on}/D) \quad (2)$$

where D denotes the number of nanodevices, R_{con}, R_{wire} and R_{on} denote the pin-to-nanowire contact resistance, the nanowire resistance and the on-state nanodevice resistance respectively, α denotes the average probability that the driving voltage to the input nanowires is high, voltage on the nanodevices is over V_t, β is the probability that the nanodevices are in the on-state, M and N denote the number of vertical and horizontal nanowires respectively.

By increasing the number of nanodevices, and vertical and horizontal nanowires in Eq. (2), the working power is increased. Since the number of required nanodevices and nanowires, in the proposed method is reduced in comparison with the methods utilized a square array of $n \times n$ nanodevices, the working power in the proposed method is considerably reduced in comparison with other methods such as methods utilized by Folling et al. [8], Likharev [21,22], and Turel et al. [41]. Fig. 7 shows an illustrative example in which the working power is shown based on the number of nanodevices D.

As it is shown in Fig. 7, by increasing the number of nanodevices, the working power in the MLP network is increased.

The leakage power is also determined based on the off-state nanodevices current as follows [9]:

$$P_{leakage} = \alpha(1-\alpha)NMV^2/\left(2R_{con} + R_{wire} + R_{off}/D\right) \quad (3)$$

where R_{off} denotes the nanodevice resistance in the off-state. In Eq. (3), the number of nanowires and nanodevices has directly effect on the leakage power. Since in the proposed method the number of nanowires and nanodevices is reduced in comparison with other MLP network implementation methods which use a square array of $n \times n$ nanodevices to determine the synaptic weight such as Folling et al. [8], Likharev [21,22], and Turel et al. [41], the leakage power in the proposed method for the MLP network implementation is also reduced in comparison with these methods.

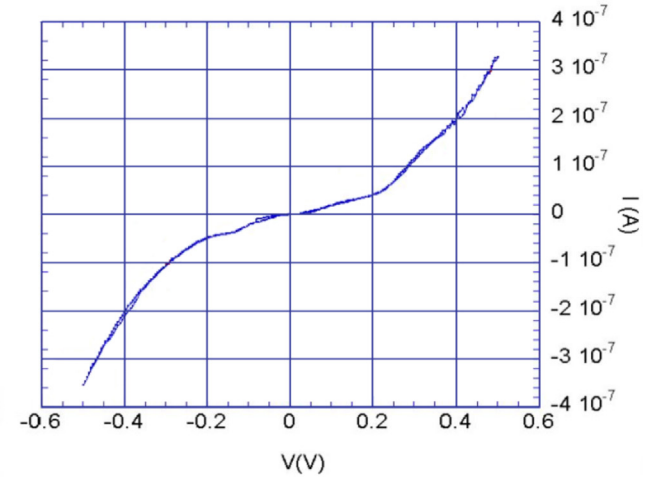
4.2. Speed evaluation

The delay time from the input pin to the output pin through the nanowires and nanodevices is determined as follows [9]:

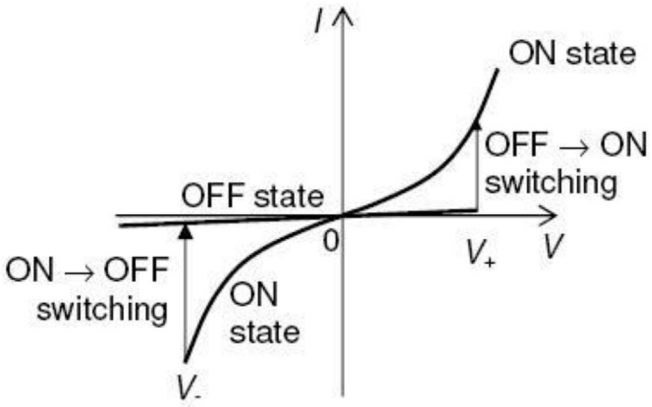
$$\tau = (2R_{con} + 1.5R_{wire} + R_{on}/D)C_{wire} \quad (4)$$

where C_{wire} denotes the nanowire capacitance.

In the proposed method for the MLP network implementations, only one nanodevice is utilized to implement each synaptic weight. So, C_{wire} is reduced in comparison with methods utilized a square array of $n \times n$ nanodevices to determine the synaptic weights. However, reducing the number of nanodevices can increase the delay time, but reducing C_{wire} can reduce the delay time. Our simulation results show that the total delay time is reduced. Therefore, the speed of the proposed method for the MLP network implementations is increased in comparison with the MLP implementation methods utilized a square array of $n \times n$ nanodevices to



(a)



(b)

Fig. 6. The *I*–*V* characteristic of a nanodevice in on-state a) simulation results b) implementation results [21].

determine the synaptic weight such as Folling et al. [8], Likharev [21,22], and Turel et al. [41].

4.3. Defect tolerance evaluation

One of the more demanding functions of MLP networks is the pattern classification. This function can be achieved after supervised training using one of established method such as error backpropagation [22]. One major concern to utilized CrossNets for pattern classification has been defect tolerance. The most numerous and significant types of fabrication-induced faults, are stuck-at-open defects in nanodevices. These defects correspond to permanently disconnected Crosspoint [6,22,33,35]. The other, less important types of defects, might include: defective nano-to-CMOS interface pins, broken or shortened nanowires, defective CMOS circuitry, and stuck-at-close defects in nanodevices [31,35]. So, the stuck-at-open defects are investigated in this paper using MNIST database [27]. In this paper, the stuck-at-open defects are assumed to be uniformly distributed. Chen et al. [2] and Strukove [35] have shown that any clustering of defects is much easier to cope with. The MNIST database is divided into two sets, one for training,

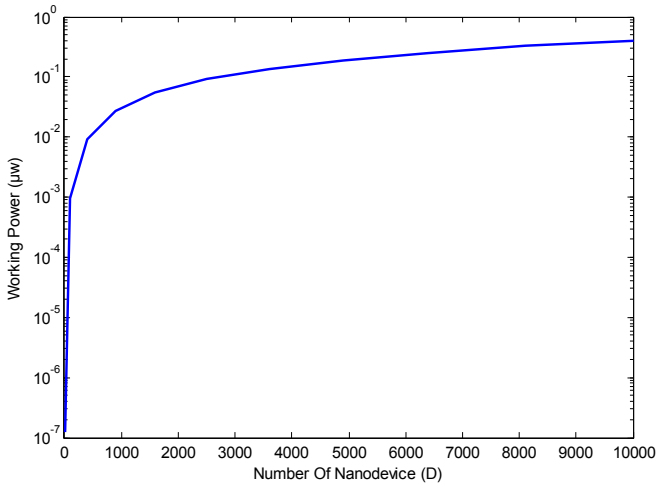


Fig. 7. An illustrative example of relationship between working power and number of nanodevices in which by increasing the number of nanodevices, the working power is increased.

60,000 images, and another for testing, 10,000 images. The training set may also divided into two sets, one for training, 50,000 images, and another for validation, 10,000 images [4]. Fig. 8 shows 200 misclassified digits utilized for network training and testing.

Data shown in Fig. 8a is a sample set of data utilized for network training and data shown in Fig. 8b is a sample set of data utilized for network testing. Each digit in Fig. 8 is shown by $28 \times 28 = 784$ pixels. As a result, a network with 784 inputs in the input layer of artificial neural network is required. In this paper, a three layers $784 \times 784 \times 10$ neurons MLP network is implemented. Bad output pixels fraction as a function of wrong nanodevices is simulated for this network using 4000 training samples and 2000 testing samples, some of them are shown in Fig. 8. The simulation results are shown in Fig. 9.

In this figure, blue line shows the results of the proposed method and red line shows the results of methods utilized a square array of $n \times n$ nanodevices. These results show that the proposed method may provide a 98% result fidelity with as many as 80% of wrong nanodevices. However, the defect tolerance of methods utilized a square array of $n \times n$ nanodevices is slightly better than the defect tolerance of our proposed method where the fraction of wrong nanodevices is less than 80%.

Based on our analysis which is shown in Fig. 9, the defect tolerance in the proposed method for the MLP network implementations is closed to methods of the MLP network implementation which used a square array of $n \times n$ nanodevices to determine the synaptic weight such as Folling et al. [8], Likharev [21,22], and Turel et al. [41].

5. Conclusion

Nanotechnology provides new opportunities for the VLSI designs. This paper investigates implementation of artificial neural networks in the nanotechnology. A novel implementation method for artificial neural network implementation in CMOL technology is presented and evaluated. The MNIST database is utilized to training and testing the proposed implementation method. Some important indices including speed, power consumption and defect tolerance are also investigated. Our analysis shows that the speed and power consumption of the proposed method for the MLP network implementations are enhanced in comparison with the MLP network implementation methods utilized a square array of $n \times n$ nanodevices to implement the synaptic weight such as

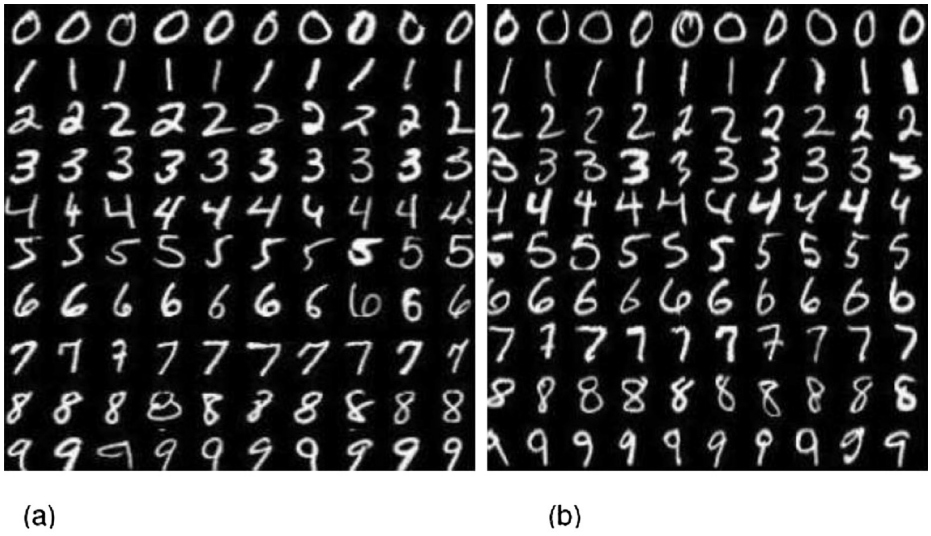


Fig. 8. Standard MNIST data a) training samples, b) testing samples.

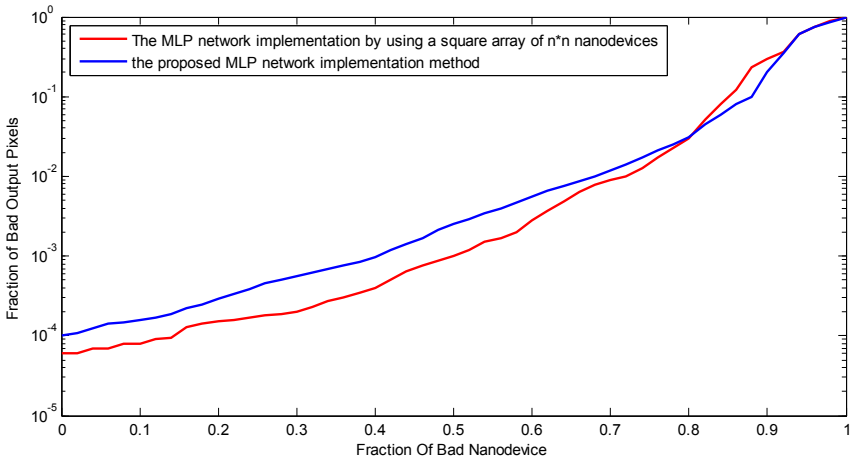


Fig. 9. Comparison of the defect tolerance of the proposed MLP network implementation method and the MLP network implementation methods utilized a square array of $n \times n$ nanodevices. Blue line shows the results of the proposed method and red line shows the results of methods utilized a square array of $n \times$ n nanodevices.

Folling et al. [8], Likharev [21,22], and Turel et al. [41] at the expense of a reasonable defect tolerance overhead. Therefore, the proposed MLP network implementation method has a huge potential to become an efficient method for implementing MLP networks. As a future works, we are planning to work with CMOS subsystem to enjoy further optimizations to have an efficient performance. In addition, we are planning to implement new machine learning requirements such as contrast normalization between layers and filter competition [17] in hybrid CMOS/nanodevice technology.

Acknowledgments

The authors are very grateful to Hadi Owlia for helpful discussions and to the anonymous reviewers for their valuable comments and suggestions.

References

[1] A. Afifi, A. Ayatollahi, F. Raissi, STDP implementation using memristive nanodevice in CMOS-Nano neuromorphic networks, IEICE Electron. Express 6 (3) (2009) 148–153.

[2] A. Chen, S. Haddad, Y.C. Wu, T.N. Fang, Z. Lan, S. Avanzino, S. Pangrle, M. Buynoski, M. Rathor, W. Cai, N. Tripsas, C. Bill, M. VanBuskirk, M. Taguchi, Non-volatile resistive switching for advanced memory applications, in: Proceedings of the IEEE International Electron Devices Meeting Technical Digest, 2005, pp. 31–33.

[3] K.T.T. Cheng, D.B. Strukov, 3D CMOS-memristor hybrid circuits: devices, integration, architecture, and applications, in: Proceedings of the ACM International Symposium on Physical Design, 2012, pp. 33–40.

[4] D.C. Cireşan, U. Meier, L.M. Gambardella, J. Schmidhuber, Deep big multilayer perceptrons for digit recognition, in: G. Montavon, G.B. Orr, K.R. Muller (Eds.), Neural Networks: Tricks of the Trade, Springer-Verlag, Berlin, Heidelberg, 2012, pp. 581–598.

[5] A. Chitra, A. Rajkumar, A study on paraphrase recognition using radial basis function neural network, IETE J. Res. 58 (1) (2012) 50–56.

[6] A. DeHon, H. Naeimi, Seven strategies for tolerating highly defective fabrication, IEEE Des. Test Comput. 22 (4) (2005) 306–315.

[7] A. DeHon, S. Hauck, Reconfigurable Computing: the Theory and Practice of FPGA-based Computation, Morgan Kaufmann, Burlington, MA, 2007.

[8] S. Folling, O. Turel, K.K. Likharev, Single-electron latching switches as nanoscale synapses, in: Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN01), 2001, pp. 216–220.

[9] C. Gao, D. Hammerstrom, Cortical models onto CMOL and CMOS – architectures and performance/price, IEEE. Trans. Circuits Syst. 54 (11) (2007) 2502–2515.

[10] M. Gholipour, N. Masoumi, Design investigation of nanoelectronic circuits using crossbar-based nanoarchitectures, Microelectron. J. 44 (3) (2013) 190–200.

[11] A.E. Hassanien, H.M. Moftah, A.T. Azar, M. Shoman, MRI breast cancer diagnosis hybrid approach using adaptive ant-based segmentation and multilayer perceptron neural networks classifier, Appl. Soft Comput. 14 (A) (2014) 62–71.
[12] J.R. Heath, P.J. Kuekes, G.S. Snider, R.S. Williams, A defect-tolerant computer architecture: opportunities for nanotechnology, Science 280 (5370) (1998) 1716–1721.
[13] Y. Ho, G. Huang, P. Li, Dynamical properties and design analysis for nonvolatile memristor memories, IEEE Trans. Circuits Syst. I 58 (4) (2011) 724–736.
[14] X. Hu, S. Duan, L. Wang, X. Liao, Memristive crossbar array with applications in image processing, Sci. China Inf. Sci. 55 (2) (2012) 461–472.
[15] M. Joodaki, Future prospect of nanoelectronic devicesSelected Advances in Nanoelectronic Devices, LNEE 175 (2013) 171–279.
[16] A. Khadem, G.A. Hossein-Zadeh, Estimation of direct nonlinear effective connectivity using information theory and multilayer perceptron, J. Neurosci. Methods 229 (2014) 53–67.
[17] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, in: Proceedings of Advances in Neural Information Processing Systems 25 (NIPS 2012), 2012, pp. 1106–1114.
[18] C. Kügeler, M. Meier, R. Rosezin, S. Gilles, R. Waser, High density 3D memory architecture based on the resistive switching effect, Solid State Electron. 53 (2009) 1287–1292.
[19] S. Lee, J.Y. Choeh, Predicting the helpfulness of online reviews using multilayer perceptron neural networks, Expert Syst. Appl. 41 (6) (2014) 3041–3046.
[20] J.H. Lee, K.K. Likharev, In situ training of CMOL crossnets, in: Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN06), 2006, pp. 5026–5034.
[21] K.K. Likharev, Electronics below 10 nm, in: Nano and Giga Challenges in Microelectronics, Elsevier, Amsterdam, 2003, pp. 27–68.
[22] K.K. Likharev, Hybrid CMOS/nanoelectronic circuit: opportunities and challenges, J. Nanoelectron. Optoelectron. 3 (3) (2008) 203–230.
[23] K.K. Likharev, Crossnets: neuromorphic hybrid CMOS/nanoelectronic networks, Sci. Adv. Mater. 3 (3) (2011) 322–331.
[24] K.K. Likharev, A. Mayr, I. Muckra, O. Turel, Crossnets: high performance neuromorphic architectures for CMOL circuits, Ann. N. Y. Acad. Sci. 1006 (2003) 146–163.
[25] K.C. Liu, W.H. Tzeng, K.M. Chang, Y.C. Chan, C.C. Kuo, C.W. Cheng, The resistive switching characteristics of a Ti/Gd_2O_3/Pt RRAM device, Microelectron. Reliab. 50 (5) (2010) 670–673.
[26] M. Masoumi, F. Raissi, M. Ahmadian, P. Keshavarzi, Design and evaluation of basic standard encryption algorithm modules using nano sized complementary metal-oxide-semiconductor-molecular circuits, Nanotechnology 17 (1) (2006) 89–99.
[27] MNIST, 2014. http://yann.lecun.com/exdb/mnist/.
[28] İ. Ovalı, A. Mavi, A study on cutting forces of austempered gray iron using artificial neural networks, Eng. Sci. Technol. Int. J. 16 (1) (2013) 1–10.
[29] H. Owlia, P. Keshavarzi, A. Rezai, A novel digital logic implementation approach on nanocrossbar arrays using memristor-based multiplexers, Microelectron. J. 45 (6) (2014) 597–603.
[30] Y. Pershin, M. Di Ventra, Practical approach to programmable analog circuits with memristors, IEEE Trans. Circuits Syst. I 57 (8) (2010) 1857–1864.
[31] U. Ruckert, H. Surmann, Tolerance of a binary associative memory toward stuck-at-faults, in: Proceedings of the International Conference on Artificial Neural Networks (ICANN-91), 1991, pp. 1195–1198.
[32] Y. Shang, W. Fei, H. Yu, Analysis and modeling of internal state variables for dynamic effects of nonvolatile memory devices, IEEE Trans. Circuits Syst. I 59 (9) (2012) 1906–1918.
[33] G. Snider, P. Kuekes, R.S. Williams, CMOS-like logic in defective, nanoscale crossbars, Nanotechnology 15 (8) (2004) 881–891.
[34] G. Snider, R. Williams, Nano/CMOS architectures using a field-programmable nanowire interconnect, Nanotechnology 18 (3) (2007) 1–11.
[35] D.B. Strukov, Digital Architectures for Hybrid CMOS/Nanodevice Circuit [Ph.D. thesis], Stony Brook University, 2006.
[36] D.B. Strukov, K.K. Likharev, CMOL: devices, circuits, and architectures, Lect. Notes Phys. 680 (2005) 447–477.
[37] D.B. Strukov, K.K. Likharev, Reconfigurable nano-crossbar architectures, in: Computational Concepts and Systems, 2012, pp. 347–454.
[38] D. Tu, M. Liu, W. Wang, S. Haruehanroengra, Three-dimensional CMOL: three-dimensional integration of CMOS/nanomaterial hybrid digital circuits, Micro Nano Lett. 2 (2) (2007) 40–45.
[39] O. Turel, K.K. Likharev, Crossnets: neuromorphic networks for nanoelectronic implementation, in: Proceedings of Joint International Conference on Artificial Neural Networks and Neural Information Processing (ICANN'2003), Springer-Verlag, Berlin, 2003, pp. 753–760.
[40] O. Turel, I. Muckra, K.K. Likharev, Possible nanoelectronic implementation of neuromorphic networks, in: Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN03), 2003, pp. 365–370.
[41] O. Turel, J. Lee, X. Ma, K.K. Likharev, Architecture for nanoelectronic implementation of artificial neural network new results, Neurocomputing 64 (2005) 271–283.
[42] R. Velo, P. López, F. Maseda, Wind speed estimation using multilayer perceptron, Energy Convers. Manage. 81 (2014) 1–9.
[43] M.A. Zidan, H.A.H. Fahmy, M.M. Hussain, K.N. Salama, Memristor-based memory: the sneak paths problem and solutions, Microelectron. J. 44 (2) (2013) 176–183.